Article

Looking Deeper into Images for Autonomous Road Weather Detection

Madiha Samo [1], Jimiama Mafeni Mase [1,*] and Grazziela Figueredo [1,*]

[1] School of Computer Science, University of Nottingham, UK

* Correspondence: Jimiama.MafeniMase1@nottingham.ac.uk (J.M.M.), g.figueredo@nottingham.ac.uk (G.F.)

Abstract: There is great interest in automatically detecting road weather and understanding its impacts on the overall safety of the transport network. This can, for example, support road condition-based maintenance or even serve as detection systems that assist safe driving during adverse climate conditions. In computer vision, previous work has demonstrated the effectiveness of deep learning in predicting weather conditions from outdoor images. However, training deep learning models to accurately predict weather conditions using real-world road-facing images is difficult due to: (1) the simultaneous occurrence of multiple weather conditions; (2) imbalanced occurrence of weather conditions throughout the year; and (3) road idiosyncrasies, such as road layouts, illumination, road objects etc. In this paper, we explore the use of focal loss function to force the learning process to focus on weather instances that are hard to learn with the objective to help address data imbalance. In addition, we explore the attention mechanism for pixel based dynamic weight adjustment to handle road idiosyncrasies using state-of-the-art vision transformer models. Experiments with a novel multi-label road weather dataset show that focal loss significantly increases the accuracy of computer vision approaches for imbalanced weather conditions. Furthermore, vision transformers outperforms current state-of-the-art convolutional neural networks in predicting weather conditions with a validation accuracy of 92% and F1-score of 81.22%, which is impressive considering the imbalanced nature of the dataset.

Keywords: Computer vision, Deep learning, Image classification, Loss functions, Vision Transformers, Weather detection

1. Introduction

Different types of weather severely affect traffic flow, driving performance, vehicle and road safety [1]. Statistics from the Federal Highway Administration show that increased amount of accidents and congestion are usually directly associated with hostile weather [2]. As a result, there is the need for advanced intelligent systems that accurately detect weather conditions and support safe driving and effective management of the transport network. Deep learning has emerged as one of the main approaches used for automatic weather recognition [3–6]. The state-of-the-art literature mostly employs convolutional neural networks (CNN), which are trained on outdoor weather images and subsequently label new images with a single weather class. This type of classification for roads, however, produces less accurate results, as multiple weather types are likely to occur simultaneously. For example, Figure 1 shows multiple weather conditions (i.e sunny and wet) present in a single scenario. Another limitation found in the current related work is that deep learning models are mostly trained on balanced and high variance weather datasets. This oversimplifies road weather conditions, which are characterised by highly imbalanced and more complex scenarios, such as road layouts, interacting elements, vehicles, people, and different illumination conditions. The representation learning therefore gets compromised, as road elements that could potentially allow for a more specific type of learning for the road problem are not included. There is also currently no research study investigating intelligent strategies for multi-label, highly imbalanced and complex road scenarios, such as dynamic pixel based weighting. This drives the motivation of this study to propose a publicly available realistic multi-label road weather dataset and employ vision transformers based on focal loss to address class imbalance and road idiosyncrasies.

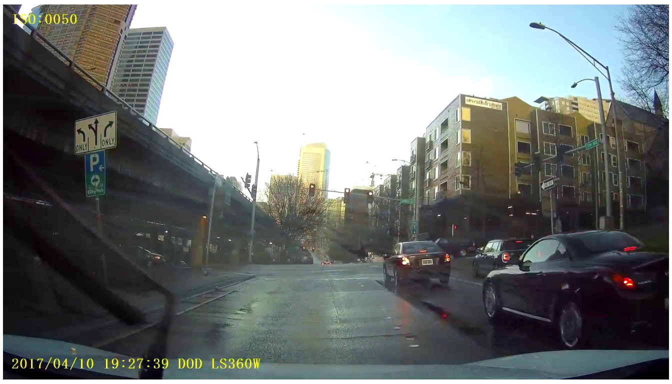


Figure 1. Multiple weather conditions (sunny and wet) existing in a single image

The main contributions of this study are:

1. A multi-label transport related dataset consisting of seven weather conditions: sunny, cloudy, foggy, rainy, wet, clear, and snowy to be used for road weather detection research.
2. Assessment of different state-of-the-art computer vision models in addressing multi-label road weather detection, using our dataset as benchmark.
3. Evaluation of the effectiveness of focal loss function to increase model accuracy for unbalanced classes and hard instances.
4. Implementing transformer vision models to assess the efficiency of their attention mechanism (assigning dynamic weights to pixels) in addressing road weather idiosyncrasies.

This paper is organised as follows, in Section 2 we review the literature on weather detection using deep learning techniques and describe the focal loss function to handle imbalanced data and difficult to classify instances. Subsequently, we provide an overview of the CNN architectures explored in this paper. Section 3 describes vision transformers in comparison to CNN networks. Section 4 introduces our novel multi-label road weather dataset, describes the vision transformer models implemented in this paper, and presents the design of our experiments and evaluation protocols. In Section 5, the results are presented along with discussion, and Section 6 concludes the paper and establishes the opportunity for future work.

2. Background

2.1. Related Work

The rapid evolution and widespread of sensors (e.g. onboard cameras) has led to large volumes of data streams constantly being generated in transportation. Deep learning approaches have emerged as suitable approaches to address big data problems as they reduce the dependency on human experts and learn high-level features from data in an incremental manner. Specifically, for weather recognition tasks, convolution neural networks have been vastly explored by many researchers.

Kang *et al.* [6] introduced a weather classification framework based on GoogleNet to recognise four weather conditions- hazy, snowy, rainy and others. Their framework was trained using the general MWI weather dataset[7] and achieved 92% accuracy. The model outperformed multiple kernel learning based approaches [7] and AlexNet CNN [8]. Similarly, An *et al.* [9] explored ResNet and Alexnet coupled with support vector machines for weather classification. The authors evaluated the models using several multi-class weather datasets. The ResNet architecture outperformed AlexNet with a classification accuracy of 92% and 88% for sunny and cloudy classes, respectively.

In Khan et al.[10], the authors developed deep learning models to recognise both weather and surface conditions based on images from road-side webcams. Their dataset consists of three weather conditions (clear, light snow and heavy snow) and three road surface conditions (dry, snowy and wet). They explored different CNN architectures including ResNet-18, GoogleNet and AlexNet, and amongst the architectures, ResNet-18 achieved the best detection accuracy with 97% for weather and 99% for road surface conditions.

Guerra *et al.* [11] introduced another multi-class Weather dataset called RFS consisting of three classes- rainy, foggy and snowy. The authors also employed ResNet architecture to achieve 80.7% accuracy on their dataset. Later, Jabeen *et al.* [12] utilized inception CNN architecture for weather detection using a new multi-class weather dataset consisting of 2000 images belonging to three classes namely, foggy, rainy and clear. Their model achieved an average of 98% accuracy for the three classes.

Zhao *et al.* [13] employed CNNs coupled with recurrent networks on a multi-label weather dataset to address the problem of more than one weather condition existing in a single image. The dataset consists of five classes including sunny, snowy, cloudy, rainy, and foggy. Their architecture achieved an average F-score of 87% for the five classes. However, the dataset used is a generalized weather dataset that is not specific to roads.

Recently, Xia *et al.* [3] explored ResNet CNNs to classify images in a multi-class weather dataset called WeatherDataset-4 into different weather conditions. WeatherDataset-4 dataset is made up of four major classes including foggy, snowy, rainy and sunny. The authors achieved an average classification accuracy of 96.03%. While Togacar *et al.* [4] employed GoogleNet and VGG16 Spiking Neural Networks (SNNs) for weather recognition. The weather dataset used by the authors consists of four classes: cloudy, rainy, sunny and sunrise. The features from GoogleNet and VGG16 are combined and trained using SNNs. The average classification result obtained with the combined CNNs and SNNs was 97.88%, which is much better than using the CNN models without SNNs.

The classification performance achieved in the above studies for weather recognition is acceptable. However, majority of the studies focused on multi-class classification which could be unrepresentative of real-world weather conditions where more than one weather condition can occur simultaneously (as shown in the sample image in Fig. 1). The few studies that employ multi-label classification [13] are either implemented on general weather dataset or fail to make their datasets available for comparison and advancement. In addition, the studies use carefully selected outdoor images which create well-balanced weather datasets. This oversimplifies the road weather detection problem, which is usually imbalanced in nature e.g. icy and snowy weather conditions rarely occur in the United Kingdom (UK). The outdoor datasets also fail to include different lighting conditions and road characteristics, making them ungeneralisable to road weather images.

We address the above limitations by proposing a multi-label weather dataset for roads to address the problem of multiple weather existing in a single frame. In addition, as the weather data is inherently unbalanced, an attention mechanism needs to be provided to address those categories that are harder to learn, as those are more likely to be extreme (rare) conditions and their misclassification by the intelligent systems should be minimised. Hence, the systematic approach followed in this study allows the model to focus more on less represented classes instead of data-dominated labels to prevent training a bias network. We also focus on feeding the model information about hard instances to avoid the gradient being outclassed by the accumulation of the losses of easy instances. Lastly, we focus on dynamically assigning weights to the pixels allowing the model to focus more on relevant features during classification which can potentially increase model's efficiency for highly complex data. Specifically, the study involves identifying the potential of adapting weighted loss and focal loss function to deal with class imbalance problems and hard-to-learn instances in the dataset. The study also involves exploring vision transformer models allowing the model to focus more on relevant pixels only. To the best of our knowledge, this study is the first attempt to recognise the potential of weighted loss, focal loss and pixel-based attention mechanism for multi-label road weather classification.

2.2. Loss functions explored in this study to deal with data predicaments

- **Class Weighted Loss Function:** The traditional cross entropy loss doesn't take into account the imbalance nature of the dataset. The inherent assumption that the data is balanced often lead to fallacious results. Since the learning becomes biased towards majority classes, the model fails to learn meaningful features to identify the minority classes. Therefore, to overcome these issues, loss function can be optimized by assigning weights such that more attention is given to minority classes during training. Weights are assigned to each class such that smaller the number of instances in a class, greater the weight assigned to that class. For each class, Weight assigned to the class = Total images in dataset/ Total images in that class. The weighted cross-entropy loss function is given by:

$$\mathcal{L} = \sum_{i=1}^{N}\sum_{c=1}^{C}\omega_c[(y_c \log(p_c) + (1y_c)\log(1p_c))] \tag{1}$$

  where L is the total loss, c represents the class, i represents the training instance, while C and N represents total number of classes and instances respectively. The $\mathrm{y_c}$ indicates the ground truth label for the class c and $\mathrm{p_c}$ is the predicted probability that the given image belongs to class c, while ω_c represents the weight of the class c.
- **Focal Loss Function:** A focal loss function is a dynamically scaled cross entropy loss function. Focal loss forces the model to focus on the hard misclassified examples during the training process[14]. For any given instance, the scaling factor of the focal loss function decays to zero as the loss decreases. Thus allowing the model to rapidly focus on hard examples instead of assigning similar weights to all the instances. Focal loss function is given by

$$\mathcal{FL}(p_o) = -\alpha_o(1-p_o)^{\gamma}\log(p_o) \tag{2}$$

  where α and γ are hyperparameters such that setting γ greater than zero reduces relative loss for examples that are easily classified. The hyperparameter $\gamma >= 0$ and its value controls the loss for easy and hard instances while α lies between [0,1] and addresses the class imbalance problem.

2.3. Deep Learning Architectures Investigated

Several state-of-the-art CNN architectures have been successfully proposed for image classification. Table 1 briefly describes the structure of state-of-the-art CNN architectures used in this study including VGG19, GoogleNet, ResNet-152, Inception-v3, and EfficientNet-B7.

Model	Author	Year	Number of Layers	Input image size
VGG19	Oxford University Researchers[15]	2014	19 layers	224 x 224
GoogleNet	Researchers at Google[16]	2015	22 layers	224 x 224
ResNet-152	He et al.[17]	2015	152 layers	224 x 224
Inception-v3	Szegedy et al.[18]	2016	48 layers	299 x 299
EfficientNet-B7	Tan et al.[19]	2019	813 layers	600 x 600

Table 1. State-of-the-art CNN models assessed in this study

3. Vision Transformers

Transformers were initially introduced for Natural Language Processing (NLP) tasks[20] while image processing tasks usually relied on convolution neural networks. Recently, transformers have been adopted for computer vision tasks [21] and they are called vision transformers. Vision transformers are similar to NLP transformers, where patches of images are used instead of sentences. Images are broken down into a series of patches and transformed into embeddings which can be easily fed into NLP transformers, similar to embeddings of words.

Conventional CNNs typically assign similar attention (weights) to all the pixels of an image during classification. As already proven in the field of NLP, introducing attention mechanisms such that higher weights are assigned to pixels of relevant information could lead to potentially better results and efficient models. Therefore, Vision Transformers (ViT) captures relationships between different parts of an image allowing the model to focus more on relevant pixels in classification problems. ViT computes relationships among pixels in small sections of the image (also known as patches) to reduce computation time instead of computing the relationship between each individual pixel. Each image is considered as a sequence of patches of pixels. However, for retaining the positional information, positional embeddings are added to the patch embeddings as shown in the figure 3. These positional embeddings are important to represent the position of features in a flattened sequence otherwise the transformer will loss information about the sequential relationships between the patches. A positional embedding (PE) matrix is used to define the relative distance of all possible pairs in the given sequence of patch embeddings and is given by the formula:

$$PE_{(pos,2_i)} = sin(pos/1000^{(2_i/d_{model})}) \tag{3}$$

$$PE_{(pos,2_i+1)} = cos(pos/1000^{(2_i/d_{model})})$$

where pos is the position of the feature in the input sequence, i is used to map column indices such that 0 $<=$ i $<= d/2$, and d is the dimension of the embedding space.

The results with the position embeddings are then fed to a transformer encoder for classification as shown in figure 2. The transformer encoder module consists of a Multi-Head Self Attention (MSA) layer and a Multi-Layer Perceptron (MLP) layer. The MSA layer splits the given input into multiple heads such that each head learn different levels of self-attention. The outputs are then further concatenated and passed through the MLP layer. The concatenated outputs from the MSA layer are normalised in the Norm layer and sent to the MLP layer for classification. The MLP layer consists of Gaussian Error Linear Unit (GELU) activation functions.

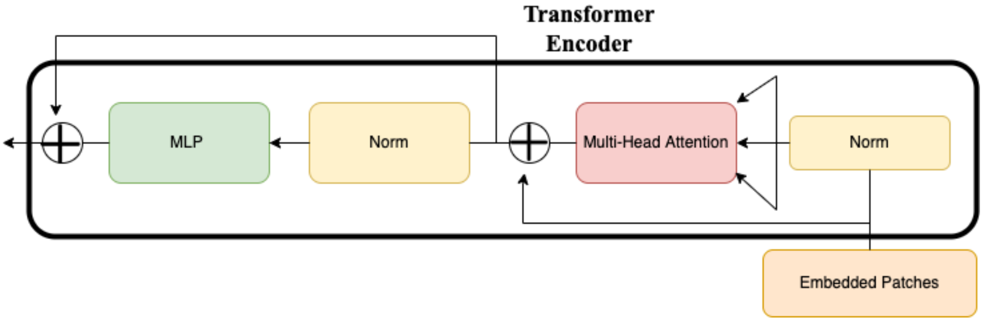


Figure 2. Overview of the Transformer Encoder.

Figure 3 shows an overview of ViT. This section concludes by explaining in more detail the attention mechanism adopted by the MSA layer.

A typical attention mechanism is based on trainable vector pairs consisting of keys and values. A set of k key vectors is packed in a matrix K (KR^{kxd}) such that the query vector (qR^d) is matched against this set of k key vectors. The matching is based on inner dot products which are then scaled and normalised. A softmax function is then applied to obtain k weights. The weighted sum of k value vectors then serve as an output of the attention. For self-attention, the vectors (Query, Key and Value) are calculated from a given set of N input vectors (i.e., patches of images) such that:

$Query = XW_q, Key = XW_k, Value = XW_v$, where W_q, W_k, and W_v are the linear transformations with the constraint $k = N$, indicating that the attention is computed between the given N input vectors.

MSA layer refers to the "h" number of self-attention functions applied to the input, as follows: $Multihead(Q, K, V) = [head1, \ldots, headh]W_0$, where W refers to the learnable parameter matrices. MSA computation is made such that query, key and value vectors are split into N vectors before applying self-attention. The self-attention process is then applied to each split vector individually. The independent attention modules are concatenated and linearly transformed.

We conclude this section by summarising the image classification process of ViT using the self-attention mechanism and encoder layer described above. Input images are split into patches of fixed sizes and multiplied with embedding matrices. Each patch is assigned a trainable positional embedding vector to remember the order of the input sequence before feeding the input to the transformer. The transformer uses constant vector size in all the layers so all the patches are flattened to map these dimensions using a trainable linear projection. Each encoder comprises two sub-layers. The first sub-layer allows the input to pass through the self-attention module while the outputs of the self-attention operation are then passed to a feed forward neural network in the second sub-layer with output neurons for classifying the images. Skip connections and layer normalisation are also incorporated in the architecture for each sublayer of the encoder.

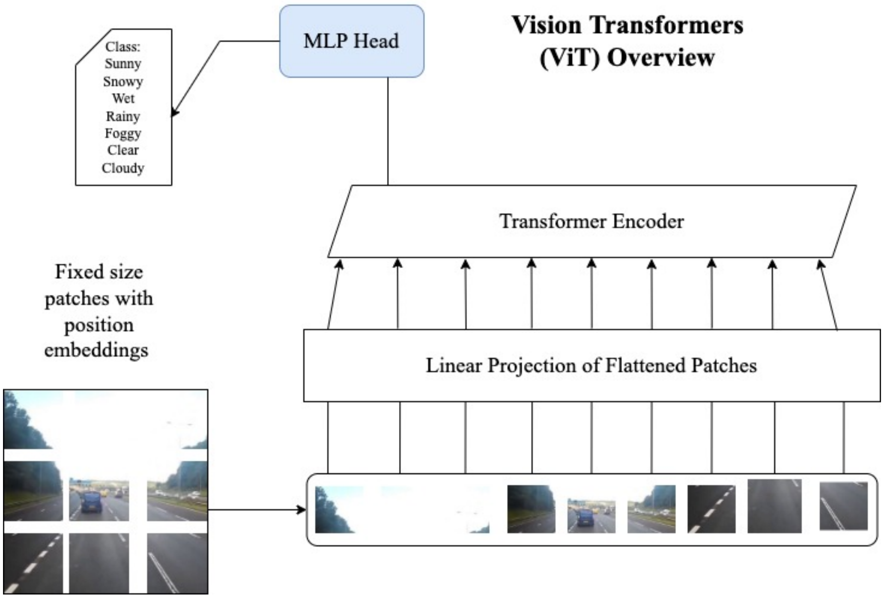

Class:
Sunny
Snowy
Wet
Rainy
Foggy
Clear
Cloudy
MLP Head
Vision Transformers (ViT) Overview
Transformer Encoder
Fixed size patches with position embeddings
Linear Projection of Flattened Patches

Figure 3. Transformer vision model architecture overview.

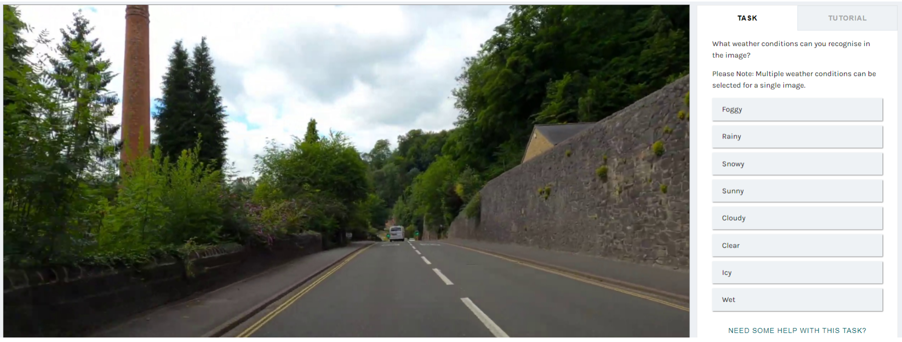
4. Experiments

4.1. Proposed Dataset Description

Due to lack of publicly available multi-label road weather dataset, we have created an open source dataset consisting of road images depicting seven classes of weather and road surface conditions i.e sunny, cloudy, foggy, rainy, wet, clear, and snowy. The images are extracted from available online videos in YouTube captured and uploaded by 'Alan

Z1000sx'[1] using a video camera mounted on the dashboard of a heavy goods vehicle completing journeys across the UK (a sample video is available at [22]). The video clips were capture different roads in UK (i.e. motorways, urban roads, rural roads, and undivided highways), different weather conditions (i.e. sunny, cloudy, foggy, rainy, wet, clear, and snowy) and different lighting conditions (i.e. sunset, sunrise, morning, afternoon, night, and evening). We downloaded 25 videos uploaded by 'Alan Z1000sx' with an average duration of 8 minutes. We developed a python script to extract images from the videos every 10 seconds. A total of 2,498 images were extracted.

To annotate the images, we utilised an online annotation platform called Zooniverse [23]. In Zooniverse, volunteers assist researchers in data annotation and pattern recognition tasks. We created a project in Zooniverse for annotating the images, uploaded the images, specified the labels, and added volunteers to our project. Zooniverse provides an easy to use interface for annotating the images as shown in Fig. 4. As shown in the figure, each image could be assigned to more than one weather condition. The annotations were carried out by two volunteers. After annotating the images, Zooniverse offers an option to export the annotations to a comma-separated values file. Table 2 shows the distribution of the images in the different weather conditions. The dataset is imbalanced with majority of the images classified as clear and sunny, while icy is the least classified as UK roads are rarely icy. Six sample images from the dataset are shown in Fig. 5 and the complete dataset is available online at [24]



What weather conditions can you recognise in the image?

Please Note: Multiple weather conditions can be selected for a single image.

Figure 4. A screenshot of using Zooniverse to annotate road weather images.

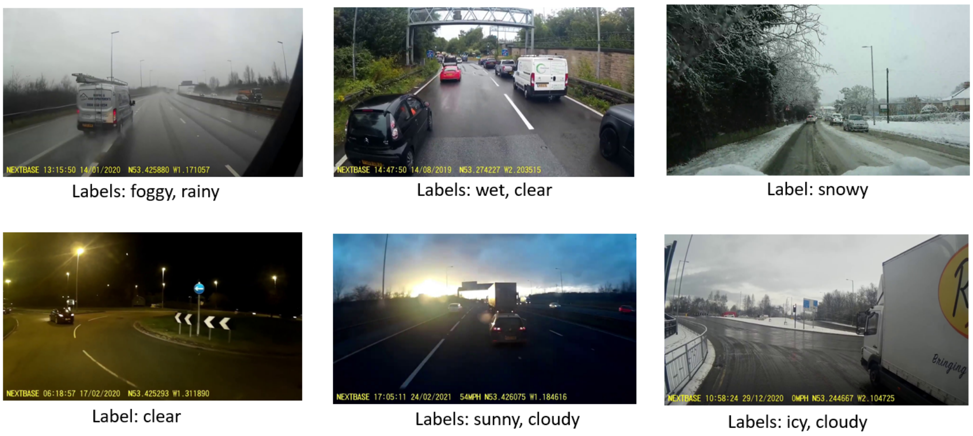
Class	Number of Instances
Sunny	1184
Clear	1299
Cloudy	626
Wet	369
Snowy	147
Rainy	84
Foggy	78

Table 2. Class distribution of the proposed road weather dataset.

4.2. Vision transformers implemented

Popular vision transformers include ViT-B and ViT-L architectures. Both architectures differ from each other with respect to the dimension of flattened patches D such that D equals 768 for ViT-B and 1024 for ViT-L. In this study, pre-trained ViT-B models are adopted as their lower dimension makes them faster to train. We employ two variants of the ViT-B model corresponding to the input patch size,including ViT-B/16 and ViT-B/32. The former refers to the input patch size of 16 x 16 whereas the latter corresponds to 32 x 32 patch size.

[1] The YouTube account that owns the road-facing videos


Labels: foggy, rainy

Labels: wet, clear

Label: snowy

Label: clear

Labels: sunny, cloudy

Labels: icy, cloudy

Figure 5. Six samples of weather images from our multi-label road weather dataset

Smaller patch sizes are resource-intensive. The models are pretrained on the 21k-ImageNet dataset which consists of 21k classes and 14 million images. Furthermore, the architecture is fine-tuned on the ILSVRC-2012 ImageNet dataset consisting of 1k classes and 1.3 million images.

4.3. Experimental Design

The training and evaluation process for the CNN architectures comprised of four stages:

1. **Stage 1:** Pre-trained the state-of-the-art CNN architectures on the ImageNet dataset.
2. **Stage 2:** Re-trained the architectures on our proposed road weather dataset using cross entropy loss function.
3. **Stage 3:** Optimise the architectures using class weighted loss function.
4. **Stage 4:** Optimise the architectures using focal loss function.
5. **Stage 5:** Pre-trained the state-of-the-art Transformer vision models on the ImageNet dataset.
6. **Stage 6:** Re-trained the architectures on our proposed multi-label road weather dataset.

In the first stage, ImageNet[25] dataset is utilized to pre-train the CNN architectures: VGG19, GoogLeNet, ResNet-152, Inception-v3 and EfficientNet-B7. We chose these architectures due to their remarkable image classification performance on the ImageNet dataset [26,27]. The images are first resized into the required image size for the CNN architectures e.g. 224 x 224 for most of the models except EfficientNet-B7 and Inception-v3 which require input size of 600x600 and 299x299 respectively. Later, the models are pre-trained by setting the 'pretrained' parameter in the models to True (in Pytorch).

In stage 2, the pre-trained models are re-trained on our proposed road weather dataset by replacing the number of outputs in the final fully connected layer of the CNN models with the number of weather classes (i.e. seven classes for our multi-label weather dataset). Only the last layers of the CNN architectures are optimised during the training process using cross entropy loss.

In the third stage, we update the cross entropy loss to incorporate the number of images in each class (i.e. class weighted loss function). This is important to reduce bias of the majority classes of imbalanced datasets by providing higher weights to images from minority classes and lower weights to images from majority classes.

In the fourth stage, focal loss function is implemented to pay more attention to classes that are harder to learn e.g. extreme (rare) weather conditions.

Since convolution neural networks assign similar weight to all the pixels during classification which might lead to inefficient results especially in a complex road image with a lot of background noise. To tackle this, in the fifth stage attention mechanism is

doi:10.20944/preprints202212.0221.v1

implemented using Vision Transformers (ViT) which are pre-trained on ImageNet dataset. In the last stage, ViT models namely, ViT-B/16 and ViT-B/32 are re-trained on the proposed road dataset for multi-label weather detection.

4.4. Evaluation Protocol

The CNN architectures were trained and evaluated using 5-fold cross validation using Pytorch programming language. The optimal learning rate for the models was set 0.001 and momentum was 0.9. A batch size of 32 and 50 epochs were utilised in training the models. It is worth mentioning that the results obtained by the training and validation set at each fold were averaged to evaluate the final performance of the models. We used the following evaluation metrics to compare the performance of models: accuracy and F1-score. Since the data is highly imbalanced, F1-score is a better metric to evaluate the models. Vision Transformer models were trained and evaluated using exactly the same hyperparameter settings and the patch size of 16 x 16 and 32 x 32 for Vit-B/16 and Vit-B/32, respectively.

5. Results and Discussion

5.1. State-of-the-art CNN models

Model	Avg Training Accuracy	Training SD	Avg Validation Accuracy	Validation SD	Avg F score	F-score SD
VGG19	84.19	0.005	85.14	0.002	58.50	0.008
GoogleNet	84.42	0.009	85.08	0.006	50.52	0.012
ResNet-152	87.58	0.003	87.73	0.005	64.22	0.014
Inception-v3	84.23	0.008	84.80	0.006	50.56	0.004
EfficientNet-B7	85.11	0.003	86.03	0.003	56.09	0.007
ViT-B/16	93.52	0.0118	**91.92**	0.0088	**81.22**	0.0182
ViT-B/32	**94.65**	0.0262	91.45	0.0065	80.48	0.0115

Table 3. Multi-label classification results for road weather detection using simple binary cross entropy loss function (best performance in bold).

Model	Avg Training Accuracy	Training SD	Avg Validation Accuracy	Validation SD	Avg F score	F-score SD
VGG19	84.48	0.002	85.35	0.005	64.21	0.015
GoogleNet	86.79	0.002	87.19	0.003	63.54	0.010
ResNet-152	88.98	0.001	88.84	0.003	71.00	0.011
Inception-v3	85.95	0.004	86.87	0.004	62.52	0.009
EfficientNet-B7	86.82	0.002	87.24	0.005	63.38	0.007
ViT-B/16	95.97	0.3579	**90.95**	0.0076	**79.18**	0.0211
ViT-B/32	**98.66**	0.0178	90.48	0.0043	77.912	0.0073

Table 4. Multi-label classification results for road weather detection using class weighted loss function to force models to handle rare weather conditions (best performance in bold).

Model	Avg Training Accuracy	Training SD	Avg Validation Accuracy	Validation SD	Avg F score	F-score SD
VGG19	83.90	0.003	84.85	0.005	66.28	0.012
GoogleNet	87.22	0.002	87.63	0.004	67.99	0.014
ResNet-152	89.44	0.004	88.71	0.007	74.40	0.010
Inception-v3	85.91	0.002	87.26	0.002	66.29	0.006
EfficientNet-B7	87.48	0.001	87.72	0.005	66.15	0.008
ViT-B/16	93.95	0.02942	**91.26**	0.0059	80.23	0.0077
ViT-B/32	**94.80**	0.3387	91.23	0.0050	**80.25**	0.0125

Table 5. Multi-label classification results for road weather detection using focal loss function to force models to handle difficult to classify weather images (best performance in bold).

Table 3 shows the multi-label classification results for the pre-trained models using binary cross entropy loss. It can be seen that ResNet-152 outperforms the other state-of-the-art CNN models in both accuracy and F1-score using our multi-label road weather dataset, followed by VGG19 and EfficientNet-B7. ResNet-152 achieves an average validation accuracy of 87.73% and F1-score of 64.22%. This result is similar to previous studies [3,9–11] where ResNet-152 showed better performance compared to other CNN architectures. However, the F1-score is low due to the imbalanced nature of the dataset.

After optimising the models using the class weighted loss function to reduce bias produced by the majority classes, we observe the classification results in Table 4.The table shows the multi-label classification results for the pre-trained models with class weighted loss function. Weights assigned to each class correspond to the *total number images* divided by *total images in that class*. It can be seen that by optimising the models with the class weighted loss function, performance has improved significantly. The best performing model, ResNet-152, now has an average F1-score of 71%. The performance of the other models have also improved with VGG19 and EfficientNet-B7 still being the second and third best models. The validation accuracy for all the models also increases.

When we focus on difficult images to classify, focal loss function is used to optimise the models. Table 5 shows that by using focal loss function, performance further improves. ResNet-152 still outperforms the other models with 74.4% F1-score. However, the best overall improvement can be seen for the model GoogleNet with a 17.74% from binary cross entropy loss to focal loss function and 4.45% increase from class weighted loss function to focal loss function. GoogleNet and Inception-v3 are now the second and third best performing models instead of VGG19 and EfficientNet-B7.

5.2. Vision Transformers

Lastly, given the highly imbalanced nature of our dataset, the results achieved so far are satisfactory. However, overcoming the limitations of CNN model, the transformer vision model further incorporates attention mechanisms to the instances forcing the model to focus on relevant pixels only. Tables 3, 4, and 5 show the results obtained from the pre-trained ViT models - ViT-B/16 and ViT-B/32. It can be seen that incorporating attention mechanisms in the architecture has significantly improved the overall accuracy as well as F-score for our multi-label road dataset. The validation accuracy achieved is 91.92% along with 81.22% F-score which outperforms all the CNN models. Nevertheless, Vision Transformers seem to be outperforming all the CNN models including the best performing focal loss based ResNet-152 with 3.72% increase in the validation accuracy and 6.82% increase in the F-score for our given dataset.

6. Conclusion

Intelligent weather detection is important to support safe driving and effective management of the transport network. Previous computer vision studies perform multi-class weather classification, which is not always appropriate and reliable for road safety, as multiple weather conditions are likely to occur simultaneously. In addition, the majority of them use balanced randomly selected outdoor images, which are unrepresentative of the real-world frequency of weather types and the unbalanced nature of road weather data. In this paper, we have introduced multi-label deep learning architectures for road weather classification i.e. VGG19, GoogleNet, ResNet-152, Inception-v3, and EfficientNet-B7. To adequately evaluate their performance, we have created a multi-label road weather dataset using naturalistic road clips captured by onboard cameras. The dataset consists of road images captured at different road types, different lighting conditions and different weather and road surface conditions. Due to the imbalanced nature of the dataset, we improved model performance using class weighted and focal loss functions to handle rare weather conditions and hard-to-classify images. Results show significant classification improvement when higher weights are assigned to rare weather conditions (class weighted loss function) e.g. snowy and icy weather, thereby, reducing overfitting on frequently occurring weather conditions such as sunny and cloudy. Additionally, further improvement is observed when the models are forced to focus more on hard-to-classify weather images (focal loss function). Furthermore, we explore attention mechanisms for pixel based dynamic weight adjustment and segmentation to improve models' performance. This is essential in separating the road layouts from the background and providing higher weights to pixels depending on the weather conditions. For example, cloudy weather can be easily recognised by analysing the background (clouds) while wet weather by analysing the road. This was achieved using transformer vision models ViT-B/16 and ViT-B/32 which outperformed all other CNN architectures. For future work, vision transformers can be implemented under different scenarios similar to CNN architectures and Grad-CAM interpretation can be implemented to observe an in-depth visual explanation to understand the learning process of ViT models under these scenarios.

Author Contributions: **Madiha Samo**- Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing - Original Draft, Writing - Review Editing, Project administration **Jimiama Mafeni Mase**- Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing - Original Draft, Writing - Review Editing **Grazziela Figueredo**- Conceptualization, Methodology, Validation, Writing - Original Draft, Writing - Review Editing, Supervision

Funding: The second author is supported by the Horizon Centre for Doctoral Training at the University of Nottingham (UKRI Grant No. EP/L015463/1) and by Microlise.

1. Mase, J.M.; Pekaslan, D.; Agrawal, U.; Mesgarpour, M.; Chapman, P.; Torres, M.T.; Figueredo, G.P. Contextual Intelligent Decisions: Expert Moderation of Machine Outputs for Fair Assessment of Commercial Driving. *arXiv preprint arXiv:2202.09816* **2022**.
2. How Do Weather Events Impact Roads?, 2015.
3. Xia, J.; Xuan, D.; Tan, L.; Xing, L. ResNet15: Weather Recognition on Traffic Road with Deep Convolutional Neural Network. *Advances in Meteorology* **2020**, *2020*.
4. Toğaçar, M.; Ergen, B.; Cömert, Z. Detection of weather images by using spiking neural networks of deep learning models. *Neural Computing and Applications* **2021**, *33*, 6147–6159.
5. Zhao, X.; Wu, C. Weather Classification Based On Convolutional Neural Networks. In Proceedings of the 2021 International Conference on Wireless Communications and Smart Grid (ICWCSG). IEEE, 2021, pp. 293–296.
6. Kang, L.W.; Chou, K.L.; Fu, R.H. Deep Learning-based weather image recognition. In Proceedings of the 2018 International Symposium on Computer, Consumer and Control (IS3C). IEEE, 2018, pp. 384–387.
7. Zhang, Z.; Ma, H. Multi-class weather classification on single images. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015, pp. 4396–4400.
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*.
9. An, J.; Chen, Y.; Shin, H. Weather classification using convolutional neural networks. In Proceedings of the 2018 International SoC Design Conference (ISOCC). IEEE, 2018, pp. 245–246.
10. Khan, M.N.; Ahmed, M.M. Weather and surface condition detection based on road-side webcams: Application of pre-trained convolutional neural network. *International Journal of Transportation Science and Technology* **2021**.
11. Guerra, J.C.V.; Khanam, Z.; Ehsan, S.; Stolkin, R.; McDonald-Maier, K. Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks. In Proceedings of the 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, 2018, pp. 305–310.
12. Jabeen, S.; Malkana, A.; Farooq, A.; Khan, U.G. Weather Classification on Roads for Drivers Assistance using Deep Transferred Features. In Proceedings of the 2019 International Conference on Frontiers of Information Technology (FIT). IEEE, 2019, pp. 221–2215.
13. Zhao, B.; Li, X.; Lu, X.; Wang, Z. A CNN-RNN architecture for multi-label weather recognition. *Neurocomputing* **2018**, *322*, 47–57.
14. Rengasamy, D.; Jafari, M.; Rothwell, B.; Chen, X.; Figueredo, G.P. Deep learning with dynamically weighted loss function for sensor-based prognostics and health management. *Sensors* **2020**, *20*, 723.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
16. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference

on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *CoRR* **2015**, *abs/1512.03385*, [1512.03385].
18. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
19. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International conference on machine learning. PMLR, 2019, pp. 6105–6114.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
21. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
22. A sample of HGV dashcam clips. *https://youtu.be/-PfIjkiDozo* **2021, last accessed on 01/10/2021**.
23. Zooniverse website. *https://www.zooniverse.org/* **2022, last accessed on 28/03/2022**.
24. Road Weather Dataset. *https://drive.google.com/file/d/1m0G_7v3U4 aPzC_BzxdyAEGV02a6Yb1CW* **2022, last accessed on 15/03/2022**.
25. Deng, J. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009* **2009**.
26. Mafeni Mase, J.; Chapman, P.; Figueredo, G.P.; Torres Torres, M. Benchmarking deep learning models for driver distraction detection. In Proceedings of the International Conference on Machine Learning, Optimization, and Data Science. Springer, 2020, pp. 103–117.
27. Pytorch. Models and pre-trained weights. *https://pytorch.org/vision/stable/models.html* **2022, last accessed on 15/03/2022**.